Story Segmentation and Topic Detection in the Broadcast News Domain

S. Dharanipragada *M. Franz* *J.S. McCarley* *S. Roukos* *T. Ward*

IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

ABSTRACT

In this paper we present algorithms for story segmentation and topic detection. Both algorithms are online algorithms and use a combination of machine learning, statistical natural language processing and information retrieval techniques. The story segmentation algorithm is a two stage algorithm that uses a decision tree based probabilistic model in the first stage and incorporates aspects of our detection system via an information-retrieval based refinement scheme in the second stage. The topic detection algorithm is an incremental clustering algorithm that employs a novel dynamic cluster-dependent similarity measure between documents and clusters. C_{seg} and topic-weighted C_{det} for these algorithms on the 1998 TDT2 Evaluation are 0.1651 and 0.0042.

1. Introduction

Automatic segmentation of a text stream, possibly the output of a speech recognizer, into its constituent stories and the detection of new topics and their clustering are important components in applications dealing with multi-media content such as browsing, searching and generating alerts. In this paper we present algorithms for story segmentation and topic detection that use a combination of machine learning, statistical natural language processing and information retrieval techniques.

The goal of a segmentation algorithm is to segment raw text, typically the output of an automatic speech recognizer (ASR), into its constituent stories. The story segmentation algorithm is a two stage algorithm. The first stage is a decision tree based probabilistic model, similar to probabilistic models described in [1], to compute the probability of a boundary at any word position given the text surrounding the position, $P(seg|text)$, while the second stage incorporates aspects of our detection system via an information-retrieval based refinement scheme.

The goal of a topic detection algorithm is to impose an organization on a collection of documents such that the underlying topical structure is exposed. The topic detection algorithm presented in this paper is an incremental clustering algorithm similar to information-retrieval based clustering techniques described in [1]. Clustering is done as soon as the document is seen i.e. without deferral. Essential to the clustering algorithm is a similarity measure between a document and a cluster and a document normalization scheme that gives the measure a natural scale and prevents large documents from dominating the clusters. We present a novel dynamic cluster-dependent similarity measure between documents and clusters.

2. Story Segmentation

2.1. System Outline

Our segmentation system is a two stage process: the first stage hypothesizes boundaries, and the second stage removes boundaries. Nonspeech events play an important role in the processing: the ASR transcript has labeled nonspeech events (such as pauses, music, etc.) along with their duration. We regard this as a crude form of sentence detection, and perform part-of-speech tagging and morphological analysis on the "sentences" of recognized speech. This is similar to the document preprocessing that we have used successfully in our information retrieval system [2]. We then model the probability of segmentation at each "sentence" boundary.

The first stage of the segmentation system uses a binary decision tree based probabilistic model to compute the probability of a boundary at every point in the ASR transcript that has been labeled a non-speech event. The features proposed for the decision tree are extracted from finite windows to the left and right of the current point. The features used by the tree are selected automatically. To determine the document boundaries from the decision tree probabilities we first find the local probability maxima in an interval of several neighboring sentence boundaries. The interval peaks are compared with a threshold value to hypothesize document boundaries. We have also incorporated a refinement stage, based on our detection metric, in which we remove posited boundaries between stories that are topically very similar.

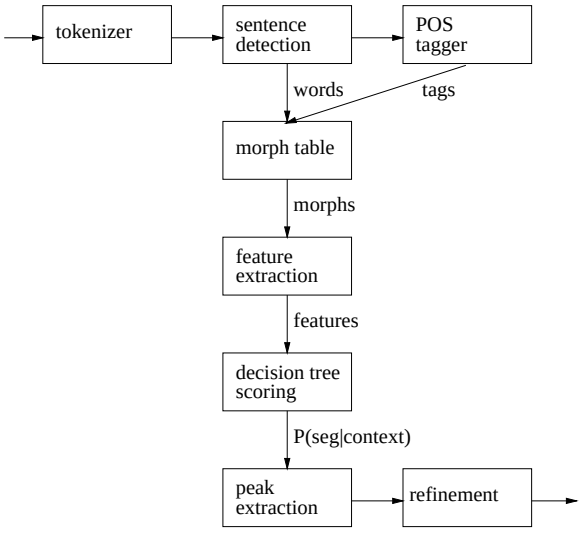

Figure 1: Components of the Segmentation System

2.2. Decision Tree Features

There are three principal types of features. First, the single most important feature is the duration of events marked as non-speech in the ASR transcript. In many cases these events are silences, which tend to be longer between stories. The next group of features is based on the presence of words and word pairs (bigrams) which are highly correlated with the document boundaries. These features, called *key* unigrams and bigrams are learned automatically from the training data based on a mutual information criterion. A related feature incorporates their average distance from the boundary. The final group of features is targeted to capture the degree of difference or similarity of the material in the window to the left and right of the current point. These features count and theshold the nouns appearing exclusively in the left and right window and in both of them - changes in subject matter are often accompanied by the introduction of many *new nouns* (i.e. nouns exclusively in the window to the right of the boundary.) A similar feature is based on left and right window semantic overlap, estimated using a symmetrized version of the Okapi formula. The top three layers of the decision tree, the feature questions, and the resulting probabilities of segmentation are shown in Figure 2.

2.3. Refinement

Our segmentation system is a two-stage process: after the story boundaries have been hypothesized, a second stage (within the deferral period) removes some of them in order to reduce the false-alarm rate. The second stage uses the document-document similarity score of our detection system (discussed below) to determine if adjacent stories are similar topically, and reject the hypothesized boundary between them. The refinement step is applied iteratively. The reduction in $P(fa)$ is of course offset by an increase $P(miss)$, but the result is a significant net gain in C_{seg}. Interestingly, the coupling between our segmentation and detection systems is more effective in the second stage than in the first stage: a similar document-similarity decision tree feature is not an important feature in the final decision trees.

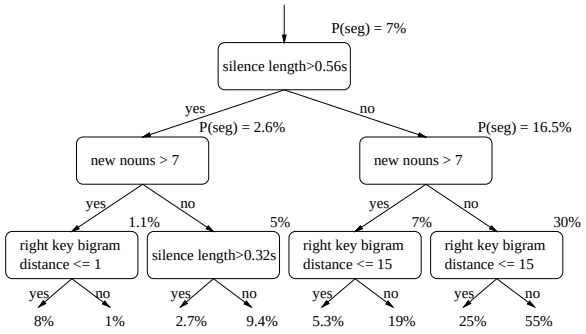

Figure 2: Top Layers of the Segmentation Decision Tree

2.4. Results

The results of the above segmentation system, using the standard DARPA metric, are summarized in Table 1. The first five lines report experiments on the development-test set. Our baseline system is as described above, without the second stage refinement. To increase the size of the available training data, we constructed additional training material by shuffling the original articles into a pseudo-random order (thus increasing the number of boundaries.) We also grew several decision trees of various depths and with differing splitting criteria and then mixed the results. Both of these yielded modest improvements. The largest single improvement was the second stage refinement (line 4) The next-to-the-last line shows the results of our submission system, incorporating all of the above experiments, on the development test set. The last line shows our official submission, trained on the first two-thirds of the data, evaluated on the evaluation set.

3. Topic Detection

3.1. System Description

The topic detection algorithm is an incremental clustering algorithm. Clustering is done as soon as the document (which has been part-of-speech tagged and morphologically analyzed, as in segmentation) is seen i.e. without deferral. Essential to the clustering algorithm is

	$P(miss)$	$P(fa)$	C_{seg}
[1]base	0.4614	0.0833	0.1967
[1]base + Monte Carlo	0.4327	0.0897	0.1926
[1]base + tree mixture	0.4688	0.0693	0.1892
[1]base + refinement	0.4166	0.0565	0.1645
[1]all, TDT2 dev. test	0.0676	0.3734	0.1593
[2]all, TDT2 eval.	0.0741	0.3776	0.1651

Table 1: Summary of segmentation experiments
[1] results on dev.-test [2] results on eval. set

a similarity measure between a document and a cluster. We measure the similarity between documents d^1 and d^2 using a symmetrized form of the Okapi formula:

$$\mathrm{Ok}\left(d^1, d^2\right) = \sum_{w \in d_1 \cap d_2} t_w^1 t_w^2 \mathrm{idf}(w, cl) \qquad (1)$$

where the term counts t_w^i of word w in document d^i have first been normalized for document length and then warped by an $x/(a+x)$ form to prevent overweighting repeated words. The cluster-dependent inverse document frequency $\mathrm{idf}(w, cl)$ is initially estimated as the standard (cluster-independent) $\mathrm{idf}_0(w)$. We represent a cluster by its centroid, i.e., the term counts t_w^{cl} of a cluster are the mean warped term counts of its constituent documents

$$t_w^{cl} = \frac{1}{|cl|} \sum_{d \in cl} t_w^d \qquad (2)$$

($|cl|$ is the number of documents belonging to cluster cl.) We note that the document-cluster score can be written as a mean of document-document scores.

We further allow the weightings of the words to vary both from cluster-to-cluster, and as the cluster evolves in time. Writing $\mathrm{idf}(w, cl) = \mathrm{idf}_0(w) + \Delta\mathrm{idf}(w, cl)$, we propose that $\Delta\mathrm{idf}(w, cl)$ should be a measure of the similarity of two sets of documents: $\mathcal{D}_w$, the set of documents that contain the word w, and the set of documents in cluster cl. In fact, we choose

$$\Delta\mathrm{idf}(w, cl) = \lambda \frac{2n_{w,cl}}{|\mathcal{D}_w||cl|} \qquad (3)$$

where $n_{w,cl}$ is the number of documents in $\mathcal{D}_w \cap cl$. which can be interpreted as a harmonic mean of a "recall" and a "precision" (if $\mathcal{D}_w$ is interpreted as a set of relevant documents, and cl as a set of retrieved documents.) Note that $\Delta\mathrm{idf}(w, cl) = 0$ if and only if $\mathcal{D}_w \cap cl$ is empty and $\Delta\mathrm{idf}(w, cl) = \lambda$ if and only if $\mathcal{D}_w = cl$.

The clustering proceeds as follows: Each document d is compared with all existing clusters. The decision to *merge*, *label* or seed a *new* cluster is accomplished by choosing the cluster cl^* that maximizes $\mathrm{Ok}(d, cl)$ and thresholding $\mathrm{Ok}(d, cl^*)$, with some exceptions. Generally, if $\mathrm{Ok}(d, cl^*) > \Theta_m$, we *merge*. If $\Theta_c \leq \mathrm{Ok}(d, cl^*) \leq \Theta_m$, we *label*. However, we only form a *new* cluster if $\mathrm{Ok}(d, cl^*) < \Theta_c$ and d contains more than 20 distinct words. We have noted that smaller documents are less stable as cluster seeds. Although our system allows Θ_m and Θ_c to be independent parameters, we have found no advantage to choosing $\Theta_m \neq \Theta_c$, except in the case of one- document clusters: then we have $\Theta_m > \Theta_c$, making it harder to merge a second document into a singleton cluster. Apparently, the C_{det} penalty for misses associated with singleton clusters is milder than the C_{det} penalty for false alarms associated with an impure initial cluster.

	trn	dev	eval
asr+nwt	0.0050	0.0021	0.0042
man_ccap+nwt	0.0047	0.0019	0.0039

Table 2: Detection results for submission system

3.2. Evaluation Results

The performance of our detection algorithm on the 1998 TDT training, development test, and evaluation sets in terms of the topic-weighted C_{det} is tabulated in Table 2 In addition to the submitted results, we also replaced the ASR transcriptions with the manually close-captioned transcriptions (man_ccap) to check the effect of speech recognition errors on our system. We observed a modest but consistent improvement in performance across all three data sets. Apparently our system is somewhat robust to speech recognition errors.

3.3. Shuffling Experiments

An event, according to the TDT definition, occurs at a specific time and a specific place. Although the topics labeled in the TDT corpus are somewhat broader (a seminal event and all directly related events and activities [3]), the TDT definition of an event may still have measurable consequences for the performance of our detection system. There are two aspects of this definition that may have an effect on our detection system. The first, which we call *temporal locality*, is that stories reporting the event tend to be localized in time. For example, 14 of the 16 stories marked YES for the Steve Fossett balloon flight topic were reported during just 3 days in January 1998. Obviously, not all topics are so well localized in time. The second aspect, which we call *temporal ordering*, is that earlier stories and later stories about an event may differ in systematic ways, as more detailed information becomes available to reporters, or the reporters assume that their readers have become familiar

with earlier reports and emphasize different aspects of the event. For example, the later stories about the Steve Fossett balloon flight are more likely to mention other balloon adventurists.

run	$P(Miss)$	$P(Fa)$	C_{det}
forward	0.1621	0.0009	0.0041
shuffled	0.1965	0.00094	0.00485
Δ shuffled	$\pm$0.0193	$\pm$0.0001	$\pm$0.00038
backward	0.1870	0.0010	0.0047

Table 3: Forward, backward, and shuffled runs; eval set

run	$P(Miss)$	$P(Fa)$	C_{det}
forward	0.0905	0.0003	0.0021
shuffled	0.1415	0.00039	0.00323
Δshuffled	$\pm$0.0313	$\pm$0.00005	$\pm$0.00063
backward	0.1635	0.0003	0.0036

Table 4: Forward, backward, and shuffled runs; dev set

run	$P(Miss)$	$P(Fa)$	C_{det}
forward	0.2210	0.0006	0.0050
shuffled	0.2600	0.0006	0.00578
Δshuffled	$\pm$0.0163	$\pm$0.0001	$\pm$0.00032
backward	0.2220	0.0007	0.0051

Table 5: Forward, backward, and shuffled runs; trn set

One reason to expect that temporal locality affects the performance of our detection system can be understood by considering the early growth of the cluster. The evolution of a cluster is more strongly influenced by articles that are added to it early than by articles that are added later. An off-topic article added early to a cluster affects all future decisions, whereas an off-topic article added later cannot retroactively affect decisions already made. Suppose that the genuine first article of a topic is taken as the seed article of the system cluster that will eventually be aligned to the topic by the scoring program. Now consider the state of the system cluster when the second topical article is encountered. If any other articles are present in the system cluster, they are necessarily off-topic, and will be scored as false-alarms. Furthermore, the system cluster will be "diluted" and it is less likely that topical articles will be merged into it. Thus misses are more likely also. If the stories reporting the event are localized in time, then the second topical article is likely to be encountered by the system soon after the seed article, and it is less likely that off-topic articles are present in the system cluster, simply because fewer off-topic articles have been presented to the system.

We can test whether temporal locality affects the performance of our detection system by shuffling the corpus into a random order and thus breaking the locality of those topics that are temporally localized. We then run our detection system on the shuffled articles. We emphasize that our detection system does not explicitly take into account the dates of the stories, even though that information is available. In fact it only knows about the order of the stories, not their absolute chronology. In Tables 3, 4, 5 we compare the forward run (with the submission system) with the 10 shuffled runs. We see that the forward run has a noticeably lower C_{det} than the mean C_{det} of the 10 shuffled runs. This difference seems to be entirely accounted for in the $P(miss)$, rather than in the false alarms. The shuffled runs have surprisingly large variance, much larger than expected based on such observations as the amount of "noise" in C_{det} during experiments to tune the thresholds.

We have also investigated the possibility that temporal ordering within a topic may affect detection performance by reversing the order of the stories and running the detection system on the reversed corpus. The results of this experiment are more ambiguous. The reverse run performed very well on the training set, but poorly on the development test and evaluation sets.

4. Conclusions

We have presented a novel approach to segmentation that couples a probabilistic model for hypothesizing segmentation with an information- retrieval approach to removing boundaries within topically homogenous material. We have also presented an information-retrieval based approach to document clustering that incorporates a novel dynamic, cluster-dependent measure of document-cluster similarity. Experiments with shuffling documents show that our detection system is sensitive to the temporal ordering inherent in the TDT definition of an event.

5. Acknowledgements

This work is supported by NIST grant no. 70NANB5H1174.

References

1. J.Allan, J.Carbonell, G.Doddington, J.Yamron, and Y.Yang, "Topic Detection and Tracking Pilot Study Final Report", in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop.* February, 1998.
2. M.Franz, J.S.McCarley, and S.Roukos, "Ad hoc and multilingual information retrieval at IBM", in *The 7th Text REtrieval Conference (TREC-7)* ed. by E.M. Voorhees and D.K.Harman.
3. "The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan", Version 3.7, Aug. 3, 1998.